

Título del video: Introducción a la programación

Textos en narración:

La programación ha estado con nosotros por un largo tiempo, y seguirá por mucho más; sin embargo, la forma en que le damos las instrucciones a la computadora ha cambiado, busca ser más sencilla para el desarrollador y así aumentar su eficiencia.

En esta Lección aprenderás sobre la evolución e historia de los lenguajes de programación, paradigmas y los lenguajes más utilizados.

Empecemos revisando qué es un lenguaje de programación.

Es una forma de comunicarnos a través de un conjunto de reglas gramaticales (tanto sintácticas como semánticas) que nos permite darle una instrucción a un ordenador, dispositivo o sistema informático para que se comporte de una cierta manera.

De acuerdo con Epitech España, la historia y evolución de los lenguajes de programación es la siguiente:

- 1801 - Tarjetas perforadas
- 1842 - Primer algoritmo informático
- 1936 - Máquina de Turing
- 1957 - Fortran
- 1959 - Cobol
- 1964 - Basic
- 1969 - Lenguaje C
- 1970 - Pascal
- 1983 - Lenguaje C++
- 1987 - Perl
- 1991 - Python

1993 - Ruby
1995 - PHP, Java y JavaScript
2001 - Lenguaje C#
2009 - Go (De Google)
2012 - Kotlin
2014 - Swift

Para conocer más de lo que sucedió en cada fecha, revisa la siguiente Lectura.

Ahora bien, ¿qué son los paradigmas de programación?

Un paradigma de programación es el estilo de la programación o desarrollo del *software*; así mismo, indica un método de trabajar y la manera como se deben estructurar y organizar las tareas que se llevarán a cabo.

De acuerdo con Martínez Canelo los tipos de paradigmas son los siguientes:

Paradigma imperativo: Son las sentencias que se definen en el código, en las cuales se indican los pasos que debe ejecutar un programa y la secuencia de éste.

Paradigma declarativo: Este paradigma usa el principio del razonamiento lógico y radica en la descripción del resultado final que se busca, es decir, con este paradigma se pretende encontrar el qué se necesita hacer y no el cómo.

Programación orientada a objetos: Permite separar los diferentes componentes de un programa, lo cual simplifica su creación, mejoras y depuración. Este tipo de programación ayuda a disminuir los errores y permite la reutilización de código.

Programación reactiva: Se basa en escuchar lo que emite un cambio o evento en el flujo de datos, los objetos reaccionan a los valores que reciben de dicho cambio.

Dentro de estos paradigmas se encuentran los siguientes enfoques:

Enfoques adicionales subordinados de paradigmas imperativos:

Programación estructurada: Este subtipo de paradigma de programación prevé el uso de bucles anidados, condicionales, subrutinas y estructuras de control. Un ejemplo de ellos es el ciclo "*do...while*" para hacer que una instrucción, de forma automática, siempre se cumpla una condición (al menos una vez).

Programación procedimental: Este subtipo de paradigma de programación desglosa algoritmos en porciones manejables, esto se denomina como procedimientos, subprogramas, rutinas o funciones según el lenguaje de programación que se esté usando.

Programación modular: Este subtipo de paradigma de programación es muy similar al enfoque de programación procedimental. En este caso el código se divide en bloques parciales independientes, lo cual ayuda a proporcionar más transparencia y facilitar la resolución de errores. Estos bloques son llamados módulos y se pueden probar por separado.

Enfoques adicionales subordinados de paradigmas declarativos:

Programación lógica: este subtipo de paradigma de programación contiene un conjunto de principios que se pueden entender como una recopilación de hechos y suposiciones. Las solicitudes se procesan de forma que el intérprete recurre a estos principios y les aplica reglas definidas previamente para alcanzar el resultado deseado. El lenguaje representativo de este subtipo de paradigma se llama Prolog.

Programación funcional: Este subtipo de paradigma de programación consta de funciones concatenadas, en la que cada parte del programa se interpreta como una función. Las funciones del programa pueden adoptar distintas estructuras. Estas funciones se pueden vincular entre sí como datos o se pueden utilizar en forma de

parámetros. Los lenguajes que se suelen usar en este subtipo de paradigma de programación son: *Lisp, Scala, Java, Kotlin*.

Según TalentCross los 12 lenguajes de programación más utilizados son:

Python: Este lenguaje de programación se utiliza en páginas web, desarrollo de *software*, ciencia de datos, machine learning e inteligencia artificial.

C: Es uno de los más antiguos. Permite crear desde sistemas operativos, pasando por hojas de cálculo, videojuegos y hasta aplicaciones.

Java: Permite la creación de módulos reutilizables que funcionan sin la necesidad de conocer su estructura interna. Con este lenguaje se pueden desarrollar aplicaciones móviles, de escritorio, web y desarrollo de juegos.

C++: Se trata de un lenguaje compilado, multiparadigma, principalmente de tipo imperativo, orientado a objetos e incluye la programación genérica y funcional. Lo que se puede desarrollar en este lenguaje son programas informáticos, sistemas operativos, desarrollo de aplicaciones móviles, videojuegos, y aplicaciones del lado de cliente y servidor.

C#: Este lenguaje de programación está orientado a objetos. Permite el desarrollo de aplicaciones web, aplicaciones móviles, desarrollo de juegos y realidad virtual, y aplicaciones para Linux y Mac.

Visual Basic: Admite una serie de elementos de lenguaje y construcción de programación comunes. Con *Visual Basic* se pueden desarrollar aplicaciones para Windows más rápidamente.

JavaScript: Se utiliza a menudo para mejorar la interactividad de una página web, permite agregar elementos dinámicos en sus páginas de destino, como gráficos animados o efectos mouseover. Lo que se puede llegar a desarrollar con este lenguaje de programación son: Desarrollo web *front-end*, juegos y aplicaciones web.

PHP: Es uno de los primeros lenguajes de *back-end*. Este lenguaje de programación ofrece los mejores *frameworks* para el desarrollo de sitios web y aplicaciones. Algunos ejemplos de estos *frameworks* son: *Laravel*, *Symfony*, *CodeIgniter* y *Zend Framework*. PHP se usa sobre todo para desarrollo web y programas informáticos.

Assembly language: Este lenguaje de programación es de muy bajo nivel, en el cual cada instrucción corresponde a una instrucción de código o máquina de computadoras. Se utiliza en los microprocesadores para comunicarse con el *hardware*, sistemas embebidos, controladores de dispositivos de *hardware*.

SQL: Se trata de un lenguaje de programación de consulta estructurado. Es el tercer lenguaje más utilizado según las estadísticas de *Stack Overflow*. Este lenguaje de programación se usa para ciencia de datos, gestión de base de datos *back-end*, herramientas de inteligencia empresarial, informe de ventas y obtención de reportes.

Go: Este lenguaje de programación fue creado para desarrollar APIs, aplicaciones de escritorio basadas en GUI y aplicaciones web. Una de sus características es la programación concurrente. Se utiliza para realizar: desarrollo web *back-end*, aplicaciones web, programación de sistemas y big data.

Swift: Es uno de los más recientes del mercado actual. Este lenguaje ofrece características modernas, como la programación concurrente y la gestión automática de la memoria. Se utiliza comúnmente para desarrollar *software* especialmente para aplicaciones de macOS e iOS.

En conclusión, como pudiste ver, los lenguajes de programación son la forma como redactamos las instrucciones a la computadora. Existe un gran número de lenguajes, por lo que, para realizar un desarrollo en específico, se debe analizar cuál lenguaje nos da las principales ventajas para el proyecto en cuestión.

La forma en que el lenguaje indica las instrucciones, se conoce como paradigmas de programación, y forma parte de las ventajas o desventajas de la elección de un lenguaje.